A Discussion of PC Platform Balance: the Intel® Pentium® 4 Processor-Based Platform Solutions

R. Scott Tetrick, Blaise Fanning, Robert Greiner, Tom Huff, Lance Hacking, David Hill, Srinivas Chennupaty, David Koufaty, Subba Palacharla, Jeff Rabe, Mike Derr
Desktop Platforms Group, Intel Corporation

Index words: Pentium® 4 processor, platform, STREAM, performance, architecture

ABSTRACT

The quest for the balanced PC platform has been with us since the advent of the Personal Computer (PC) in 1982. The basic PC has been dramatically successful, such that a 200-fold increase in computing power has been required. At the same time, the platform has had to improve its capabilities to support these uses, and provide growth for new applications. To restore balance in the platform, processor buses, memory interfaces, and advanced platform capabilities must keep pace with, and even lead, advancements in the processor. This paper describes the technological advances made in the development of the first platform for the Intel® Pentium® 4 processor. After a brief look at PC platform partitioning through the years, the platform partitioning developed for the first Pentium 4 processor platform is discussed, beginning with an understanding of the performance of the processor, and how that is dependent on aspects of the platform. We then present two primary platform improvements for the high-performance PC platform: the 400MHz system bus and the Intel® 850 Memory Controller Hub. After providing an understanding of these platform advancements, we show how these two improvements together complement the Pentium 4 processor computational capabilities by concentrating on results obtained in the standard *SPEC CPU2000*[1] and *STREAM*[2] benchmarks. This combines the high-speed processing of the Intel Pentium 4 processor with platform improvements to provide a dramatic increase in overall performance.

[1]The next-generation industry-standardized CPU-intensive benchmark suite. SPEC designed CPU2000 to provide a comparative measure of compute-intensive performance across the widest practical range of hardware.

[2]A simple synthetic benchmark that measures sustainable memory bandwidth and the corresponding computation rate for simple vector kernels.

INTRODUCTION

With the first IBM Personal Computer (PC) in 1982, the basic platform architecture of today's PC platform was established. Improvements to this platform were necessitated by improvements to the microprocessor, as dictated by Moore's Law. While some of these improvements were small, others were major, moving also at the pace of Moore's Law. In this paper, we first describe the substantial platform changes that allow us today to support the latest Intel Pentium 4 processor.

In order to understand the impact of these improvements on the Pentium 4 processor platforms, we need to see how these improvements impact the performance of the applications. Previous discussions of the performance of the Pentium 4 processor have described the processor performance as the product of two values, frequency and Instructions Per Clock (IPC). While the increase in frequency is strictly due to advances in microarchitecture and an improved silicon process, the IPC value has a strong platform-level component as well.

In this paper, we highlight two platform advances in the Pentium 4 processor. First, the 400MHz system bus of the Pentium 4 processor, responsible for all data into and out of the processor, is discussed. This high-bandwidth connection provides the necessary throughput for today's performance applications, with headroom for new applications in the coming years. Second, the system bus advanced protocol is discussed. This improved parallelism is best shown by a careful examination of the 82850 Memory Controller Hub. The 82850 is responsible for balancing the bandwidth demands of graphics, I/O, and the processor with two channels of Direct RDRAM.

An obvious question at this point is whether these platform improvements are necessary. We present case studies to show that these platform improvements add to bottom-line performance. This is represented graphically to show how the platform's failure to meet the peak bandwidth demand of the application impacts the overall performance of the microprocessor. For many applications, even average bandwidths can show that the

high-performance aspects of the platform of the Pentium 4 processor are required. Finally, we carefully examine the *STREAM* benchmark, which combines the high-performance computational capabilities with the high bandwidth supplied by the platform of Intel's Pentium 4 processor.

THE ARCHITECTURE OF THE FIRST PERSONAL COMPUTERS

In 1982, the first IBM Personal Computer (PC) was introduced. This relatively simple design has held steady for all PC platforms since then. The processor was the Intel® 8088, running at a clock rate of about 5MHz and providing a processor bus connection of about 1.2 megabytes/second. Figure 1 shows a simplified block diagram of the early IBM PC. Moreover, it should be noted that all bandwidth in the system was routed through this processor bus. The bandwidth of this bus provided a ceiling for all the graphics and the I/O in the system.

Contrast this to the 1995 version of the PC architecture, which shows how much the platform architecture changed in just five years. The processor bus had improved 500-fold to over 500 megabytes/second. Moreover, a major platform improvement was made with the addition of the Advanced Graphics Port (AGP) interface. This repartitioning of the platform removed the graphics bandwidth from the I/O interface. This greatly improved the platform capabilities, but complicated the function of the chipset, the 82440BX.

The 2000 PC block diagram shows the platform of the Pentium 4 processor. As you can see, there have been the usual improvements to the interfaces, but a continued repartitioning of the platform. This repartitioning represents a continued improvement in the platform, as a result of advanced speeds and feeds. The growth requirements of the platform are shown in Table 1. We explore the reasons for these platform changes and how they contribute to improved platform performance.

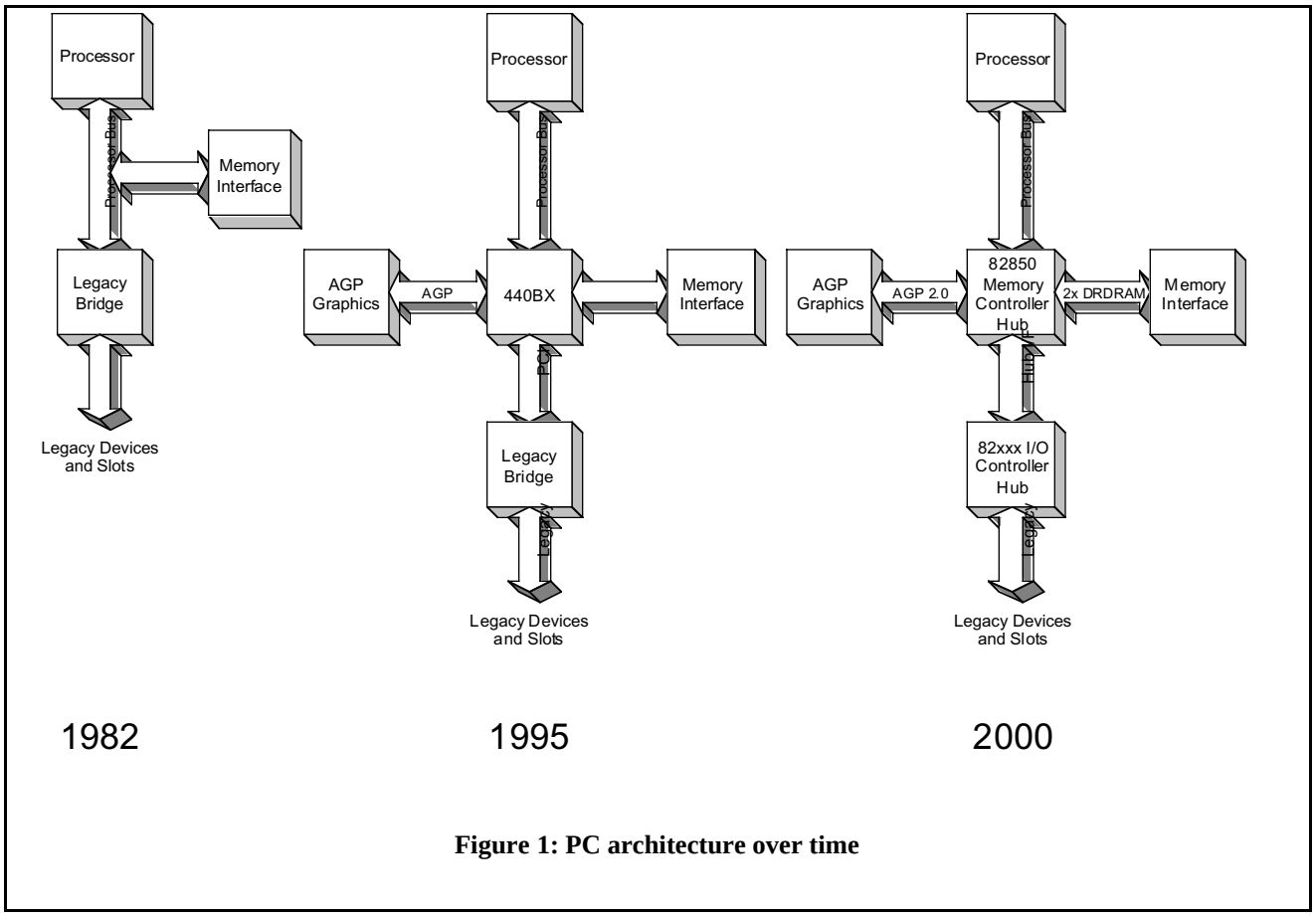

Figure 1: PC architecture over time

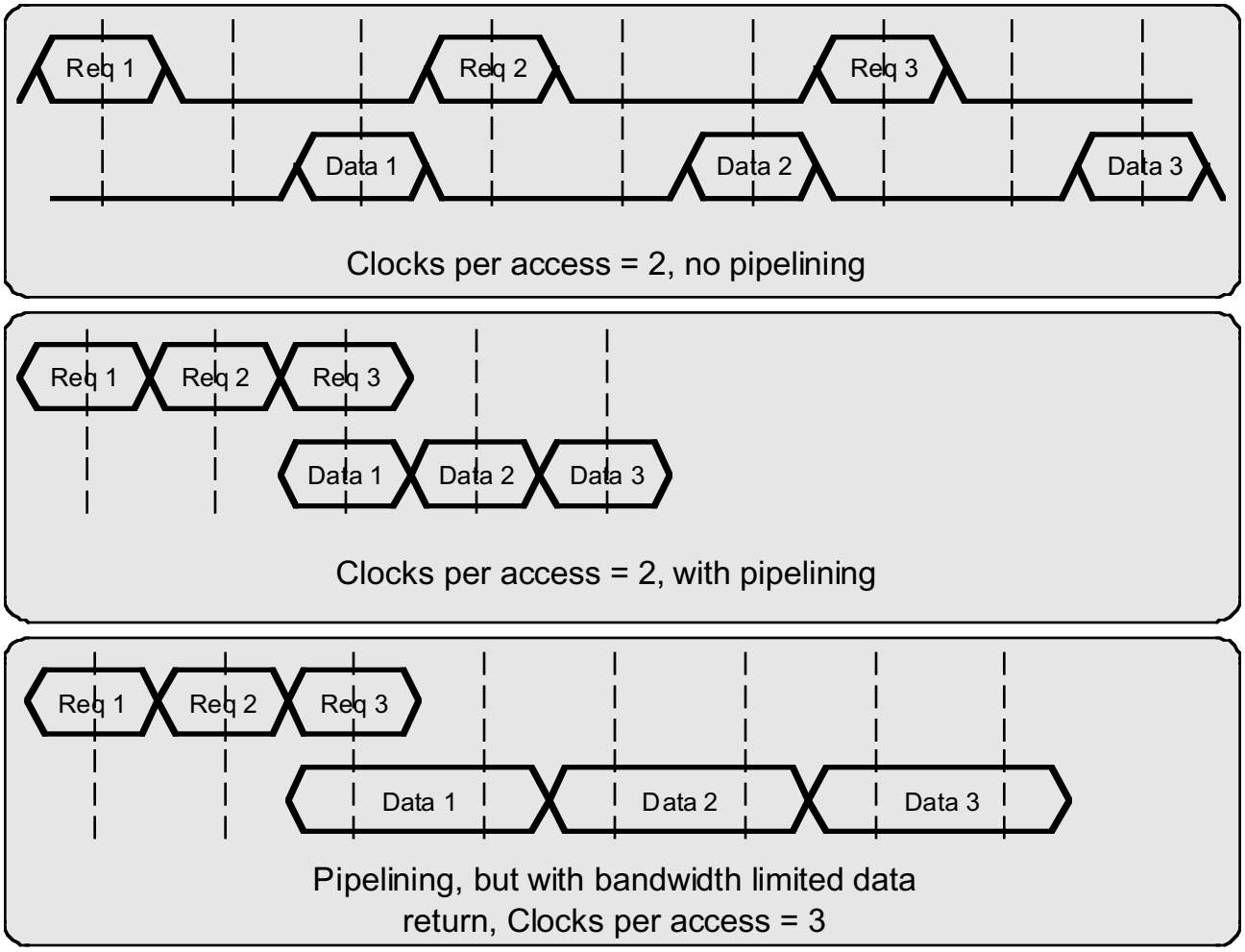

Figure 2: System bus impacts to clocks per access

Platform Interface	1982	1995	2000
Processor Bus Bandwidth	1.2 MB/s	533 MB/s	3200 MB/s
Graphics Bandwidth	N/A	533 MB/s	1066 MB/s
I/O Connection	N/A	133 MB/s	533 MB/s
I/O Bus	1.2 MB/s	Same as I/O Connection	133 MB/s

Table 1: Bandwidth increases over time

How the Platform Contributes to Performance

Before looking at results, let us take a look at the theoretical basis for the impact of the platform on performance. Fundamentally, the time to run an application is

*Execution Time = Instructions * CPI/Frequency*

Where:

Execution Time = application run time
Instructions = number of instructions
CPI = clocks per instruction
Frequency = CPU core frequency

CPI varies with the application, cache hierarchy, and I/O use. It can be further defined as

$$CPI = CPI_{base} + \sum_i P_i C_i$$

CPI_{base} is a figure of merit for the application and the core processor microarchitecture. P_i is the probability of event i occurring which has an additional latency in clocks of C_i. C_i values may be very large; for example, if the core clock is 1GHz and the average memory access is 100 ns, C_i is 100 clocks. Unfortunately, C_i for a fixed latency device varies with frequency (doubling our core frequency in the example above also doubles our C_i to 200 clocks). C_i can also be dependent on system state, the ability of the platform to support pipelining, and the queuing impact due to demanded bandwidth mismatches. Examples of these C_i cases are shown in Figure 2.

The examples presented in Figure 2 clearly lead us to the following conclusions, as we want to scale to high frequencies in the processor.

- The connection to the processor must be highly pipelined to improve bus efficiency and avoid queuing requests.
- The bandwidth to the processor must closely match the request rate to queuing responses.

HIGH-PERFORMANCE INTERFACE TO THE PROCESSOR

The *STREAM* benchmark, along with various *SPEC* benchmarks, is used to show how the platform impacts the overall performance of the system.

The *STREAM* benchmark is a simple synthetic benchmark that measures sustainable memory bandwidth and the corresponding computation rate for simple vector kernels. It represents a balance between memory bandwidth and floating-point operations. While the benchmark is artificial, it is similar to a number of applications of interest. Applications with streaming data, such as video editing, format conversion of audio and video, and encryption primitives all have behaviors common to those of the *STREAM* benchmark.

PLATFORM IMPROVEMENTS TO KEEP PACE WITH PROCESSOR

As processor clock speeds continue to grow exponentially, the system memory bandwidth required keeping these processors busy doing useful work grows as well. Ideally, when CPU frequency is doubled, application execution time should be halved. On a given processor architecture with a fixed cache size and caching strategy, the instantaneous system memory bandwidths would double, since the execution engine would still require access to all of the same pieces of memory data. It would, however, require access to them in half of the time required by the slower processor.

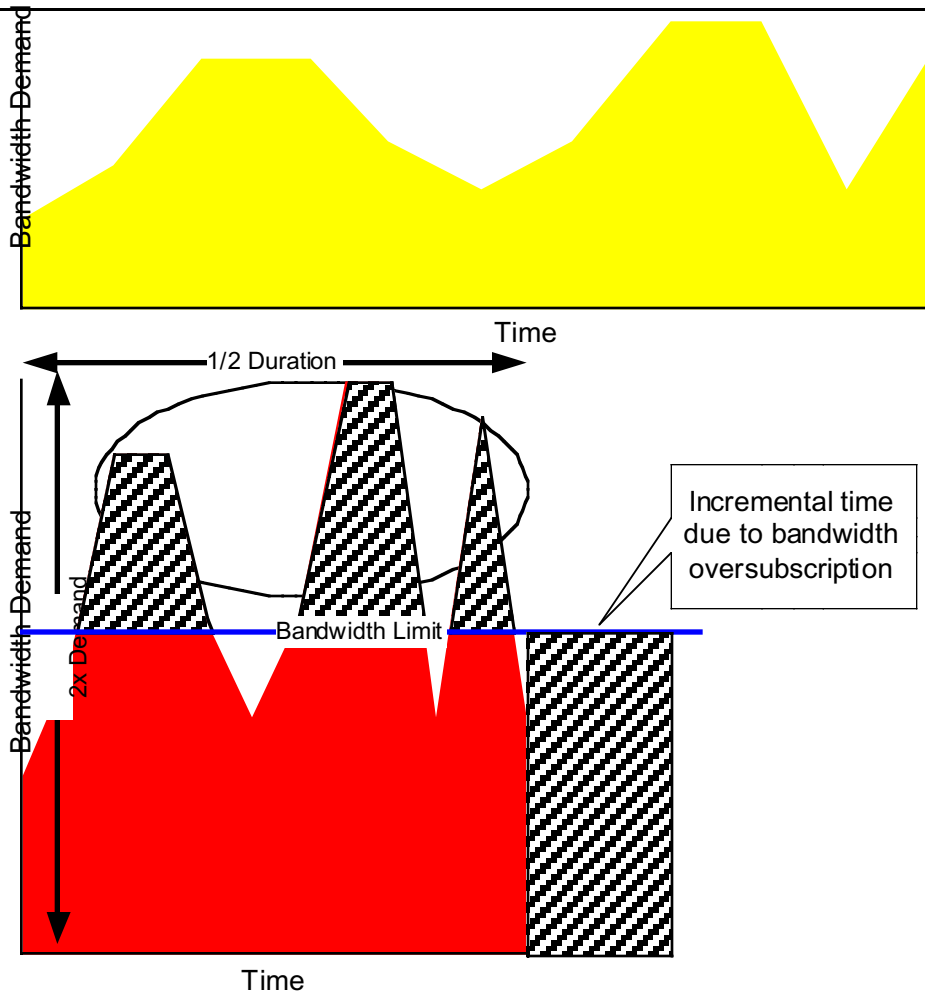

Figure 3: Frequency scaling and memory limitations

System memory application bandwidth vs. application execution time is plotted in Figure 3. The yellow area of the graph indicates the memory bandwidth demanded over time of the applications. If bandwidth is unconstrained, the processor will request memory twice as fast and obtain twice the bandwidth, and scale perfectly. In other words, it completes the application in one-half the time. However, let us assume that the memory bandwidth is constrained as indicated by the dark blue band. The increased memory demand can only be met by an increase in the execution time, as shown in the shaded area. Due to the limited memory bandwidth, the processor that is twice as fast yields less than the 2x speedup due to the finite memory bandwidth. During this process, the area under the bandwidth profile remains constant. Since the application requires its data more quickly, the bandwidth required to satisfy processor requests increases.

It is important to note that this demand profile for memory bandwidth is application specific. Many applications demand memory resources at a fairly low rate, and the platform can accommodate the processor speedup without degradation. Examples of applications that do not demand high bandwidths include word processing and presentation software, such as those components of the SYSmark 2000[3] benchmark.

Systems whose memory and bus implementations limit the amount of bandwidth available to the system execution engines decrease the ability of applications to scale with processor frequency increases. A memory

[3] An application-based benchmark that reflects today's leading-edge software applications for Internet Content Creation and Office Productivity. SYSmark is a registered trademark of The Business Applications Performance Corporation.

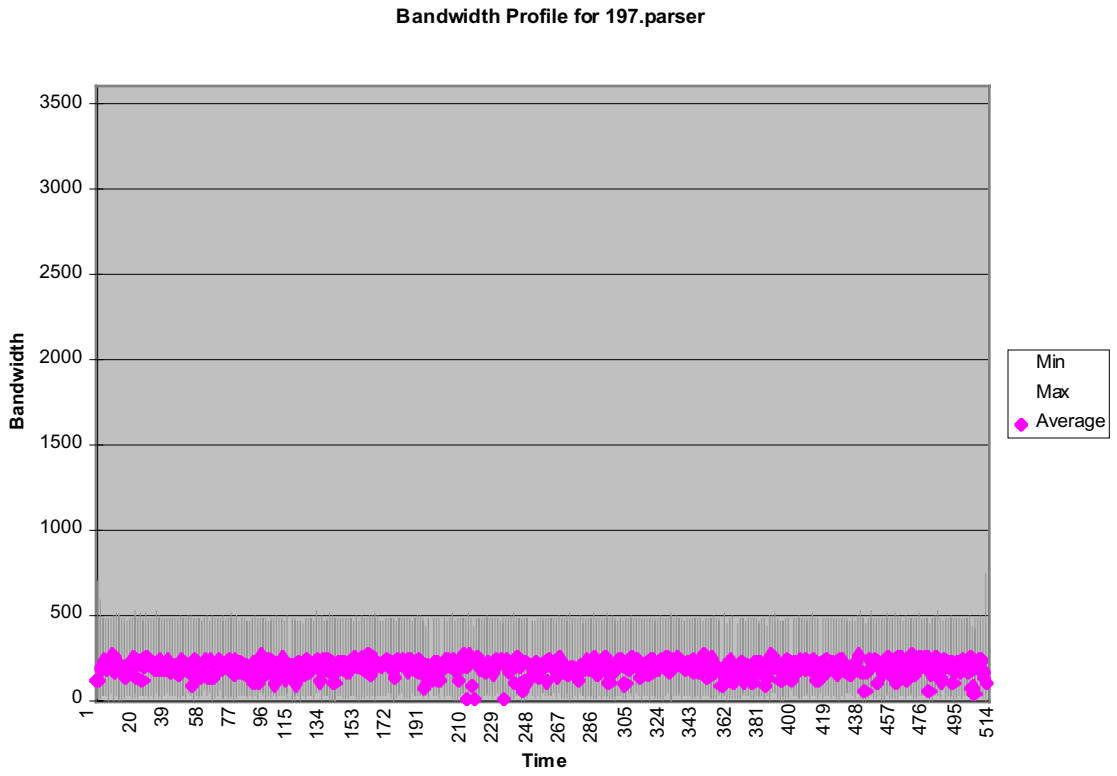

Figure 4: Pentium® III processor 197.parser bandwidth profile

subsystem implemented with a 64-bit PC133 SDRAM or with a 133MHz Pentium® III processor bus would limit the ability of the above example application to scale linearly with processor frequency. Such memory systems and processor buses can ideally deliver only 1.066GB/sec. As a result of this "ceiling", the CPU is forced to wait for some of the pieces of data that it requires during the time the bandwidth limit is exceeded, as indicated by the shaded areas in Figure 3. While the processor is waiting, it is doing no useful work and is not contributing to increased application speed.

In order to analyze the effects on the system of limited and expanded bandwidth, measurements were taken on a system with a Pentium III processor running different components of the *SPEC* CPU benchmark suite. As one might expect, there was wide variation in the amount and profile of memory bandwidth required by the discrete benchmark components. However, the benchmarks could basically be divided into two categories: benchmarks that were already bandwidth limited or nearly bandwidth limited and benchmark components that required very little bandwidth and would scale well even without the bus and system enhancements brought by the Pentium 4 processor, the Intel850, and RDRAM memory.

The first class of benchmarks, those with low-bandwidth requirements in the Pentium III processor generation, included a number of the integer benchmarks, such as the 197.parser and the 168.wupwise. Their execution profiles on a 1GHz platform with a Pentium III processor, using an Intel840 chipset, showed fairly low system bandwidth consumption. The bandwidth results for the 197.parser are illustrated Figure 4. Note that Figure 4 illustrates the average bandwidth consumption in a 1-second slice using a solid dot, while the high- and low-bandwidth limits during that same slice are shown with a gray line.

The bandwidth profile for the 197.parser demonstrates a fairly consistent average bandwidth requirement of around 250MB/sec, over the duration of the benchmark execution time. During this time, the peak data requirements stay below 500MB/sec. As a result, memory and processor bus implementations that limit processor data to around 1GB/sec should not limit even a 2GHz execution engine.

The other class of applications found in the *SPEC* CPU benchmark suite are those that are clearly bandwidth limited on the Pentium III processor platforms. Floating-point applications such as the 179.art, the 171.swim, and the 172.mgrid illustrate the problem most vividly, although integer applications such as the 181.mcf also

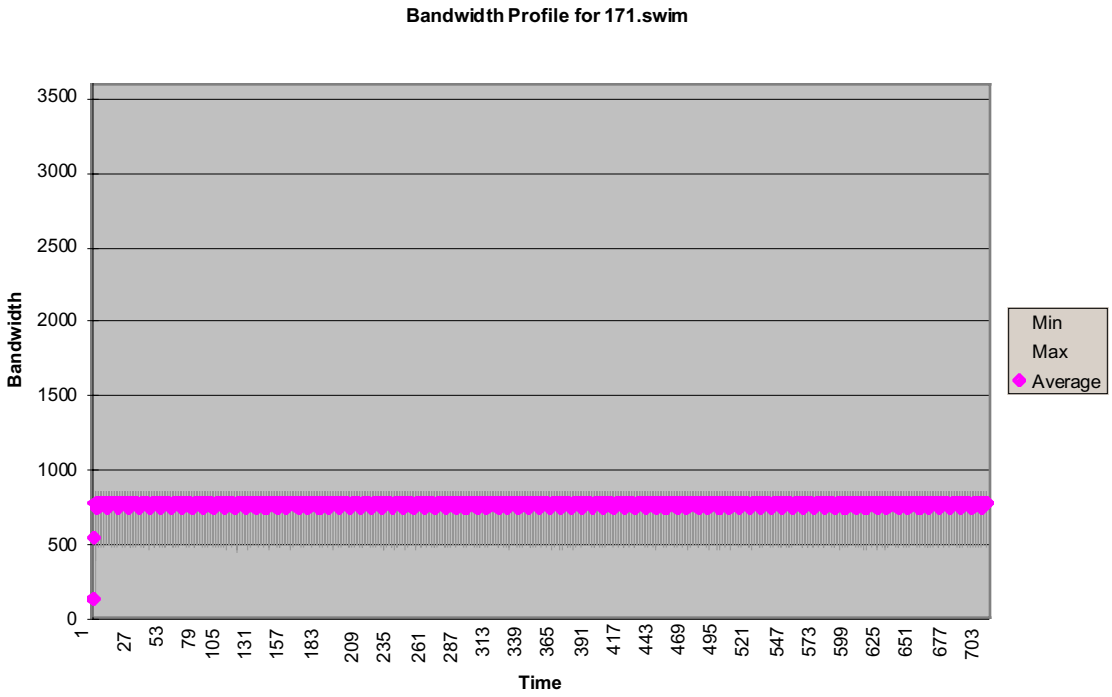

Figure 5: Pentium® III processor bandwidth profile for 171.swim

demonstrate serious bandwidth "hunger." Figure 5 illustrates the bandwidth profile for the 171.swim.

The graph clearly illustrates the bandwidth limitation of the existing Pentium III processor bus, since its average processor bus utilization is around 800MB/sec or 80% of the ideal processor bus bandwidth. If processor frequency scaling is to return benefits on this application, the bus and memory system bandwidth capabilities must clearly be increased. Existing structures do not support application runtime scaling, and applications such as the 171.swim executing on Pentium III processor platforms clearly fall into the class of being "bandwidth-limited."

Given vectors **a, b** and **c**, and scalar q, the *STREAM* benchmark measures the memory bandwidths for the following operations.

Function	Vector Operations
Copy	$\mathbf{a} = \mathbf{b}$
Scale	$\mathbf{a} = q\mathbf{b}$
Sum	$\mathbf{a} = \mathbf{b} + \mathbf{c}$
Triad	$\mathbf{a} = q\mathbf{b} + \mathbf{c}$

The sizes of the arrays are set much larger than processor cache sizes to guarantee memory is exercised. By performing both floating-point and bus operations, platform balance can be assessed. *STREAM* operations are typical for a number of new application classes, where streaming data are required to be delivered to the processor, computed, and delivered to a peripheral at high speed. This new workload is required in such things as speech recognition, video editing, and Internet servers with streaming datatypes.

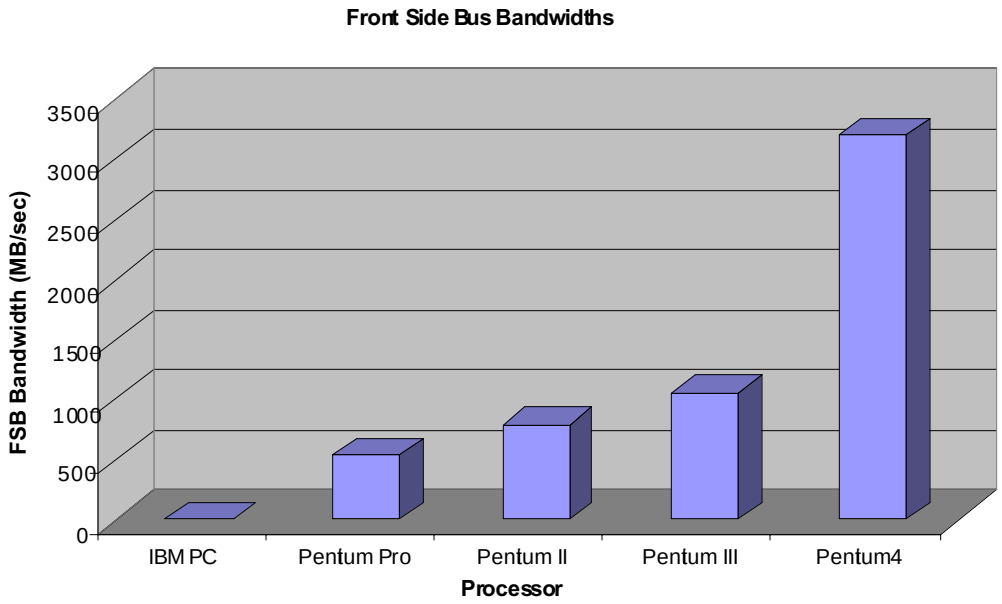

Figure 6: PC processor system bus bandwidths

PLATFORM IMPROVEMENTS DELIVER PERFORMANCE

As we have seen in the previous analysis of benchmarks, it is clear that system bus bandwidth is a limiter for the performance of the platform when high bandwidth is required by the application. This was foremost in the minds of the designers of the Intel Pentium 4 processor when developing the system bus. The system bus used in the Pentium 4 processor delivers unprecedented bandwidth for the PC platform, as can be seen in Figure 6. In addition, the system bus protocol has been improved to allow more deeply pipelined operations, memory prefetching, and glueless multiprocessing.

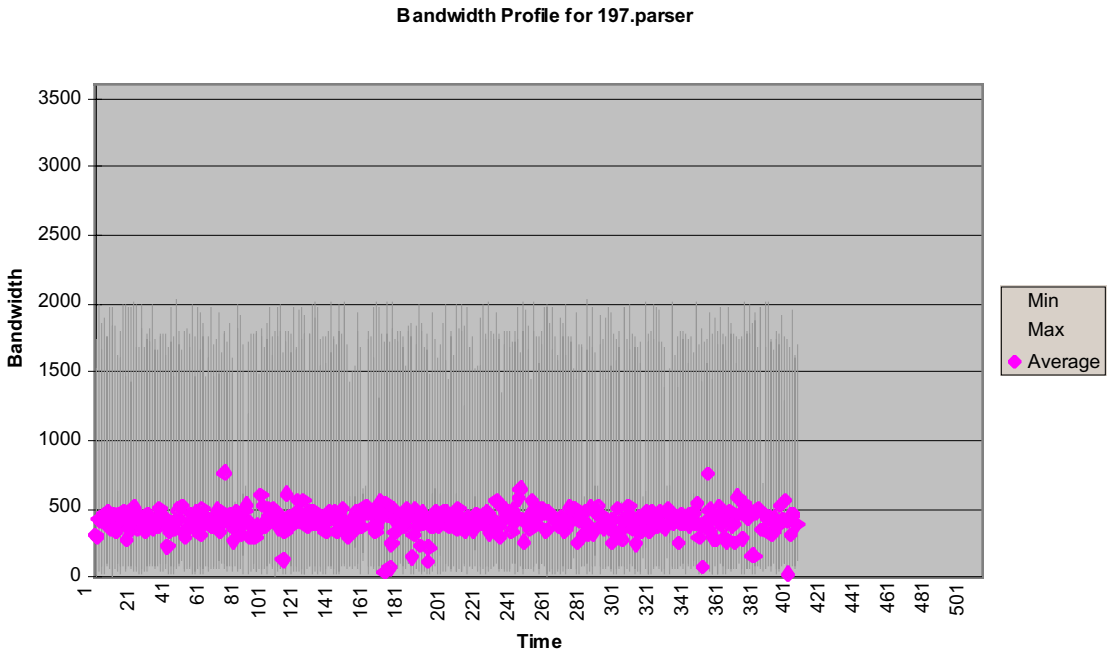

Figure 7: Pentium® 4 processor 197. parser bandwidth profile

The elevated execution speed and processor bus transfer rate of the Pentium 4 processor offer the opportunity for greatly enhanced application performance. This is illustrated by its performance on some benchmarks. To reiterate, one class of benchmarks would see reduced execution time solely because of processor frequency increases. On the other hand, different benchmarks might have found their achievable performance limited by the bandwidth on the processor bus without the advantage of the enhanced bus speed of the Pentium 4 processor. The first class of applications is typified by the 197.parser component of SpecINT, whose average and extreme bandwidths on a 1.4GHz Pentium 4 processor are plotted in Figure 7.

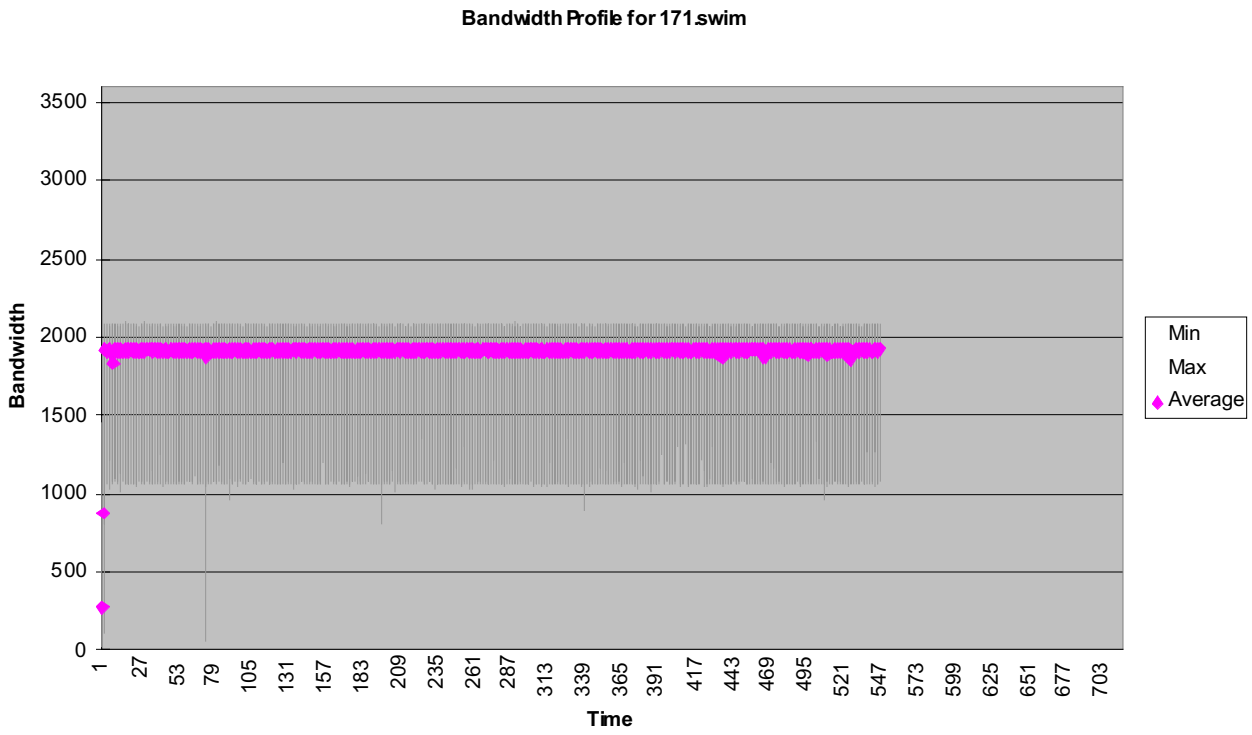

Figure 8: Pentium® 4 processor 171.swim profile

Note that the bandwidth required by the 197.parser increases dramatically from the Pentium III to the Pentium 4 processor. Where the average bandwidth of the Pentium III processor hovered around 300MB/sec, the bandwidth of the Pentium 4 processor requires around 400MB/sec during its reduced execution time.

Applications that were limited by the Pentium III processor bus experience the greatest speedup. 171.swim was shown in Figure 5 to be demanding memory bandwidth nearly equal to that of the Pentium III processor bus bandwidth. The corresponding results on a Pentium 4 system are shown in Figure 8. The bandwidth demanded by the Pentium 4 processor is nearly doubled over that of the Pentium® II processor. Since the Pentium 4 processor platform is able to satisfy the higher demand, the execution time of 171.swim is greatly reduced.

Note that systems with Pentium III processors simply did not support this high-bandwidth rate. Their maximum theoretical data transfer rate peaked at 1066MB/sec on a 133MHz bus.

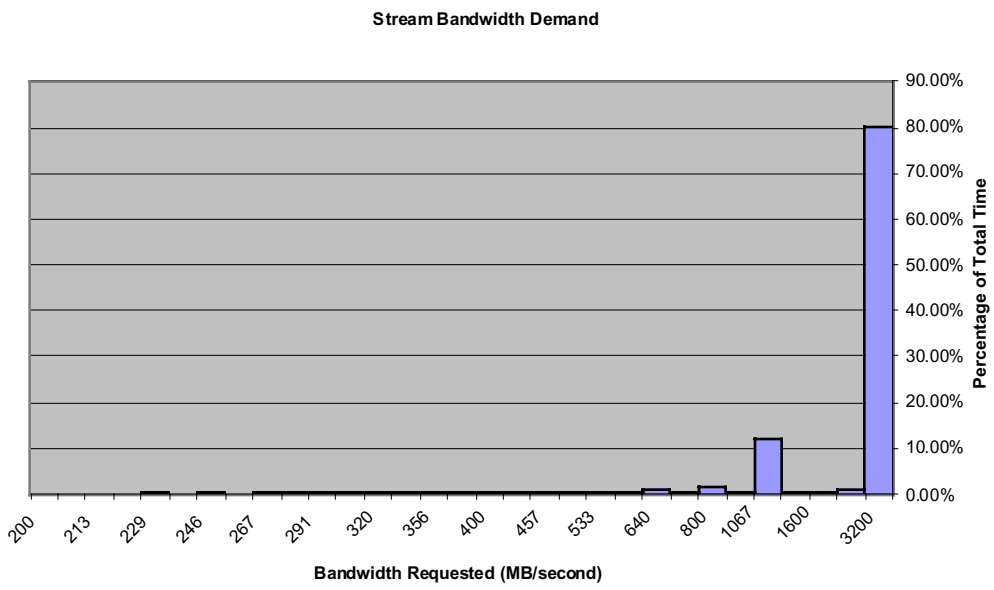

Figure 9: *STREAM* memory bandwidth demands

The analysis of the *STREAM* results is similar, but presented in a different format. Figure 9 shows that 80% of the memory requests are delivered at a 3.2GB/sec data rate. Recall too that this must be matched by the floating-point processing capability of the processor to sustain this level of performance. The high-performance floating - point unit of the Pentium 4 processor more than matches the computation task. As was the case with the *SPEC* trace results, any platform incapable of supporting the requested memory bandwidth of *STREAM* will have lower benchmark results. It should also be noted that *STREAM* analysis shows the parallelism of the Pentium 4 processor system bus, since the average number of outstanding memory requests over the duration of the benchmark is almost six, indicating that the 82850 memory controller is simultaneously processing that number of memory requests at all times. Figure 10 shows the *STREAM* results for the Pentium III processor at 1GHz and the Pentium 4 processor at 1.5GHz, demonstrating the advantages of both the high-performance execution engine of the Pentium 4 processor and the platform improvements of the system bus and memory controller.

It should be noted that this methodology is independent of the memory technology. The two technologies presented here, SDRAM or PC133 and RDRAM merely represent two different technologies that are able to provide different memory bandwidths. This analysis could equally well be applied to different memory technologies, such as DDR SDRAM.

Should the memory demand of the application be unmet by the platform, the microarchitecture of the Pentium 4 processor can still compensate to some degree. Prefetching of memory requests allows the memory controller to more efficiently serve the requests. The high-performance pipeline of the Pentium 4 processor allows much higher processor frequencies to improve performance as the processor frequency increases.

CONCLUSION

It is not sufficient merely for the microprocessor to advance down its Moore's Law trajectory. In this paper, we have discussed the need for the platform components to move in concert with the microprocessors' technological advances. While there have been significant improvements in platform capabilities over time, the platform of the Intel Pentium 4 processor provides a significant leap forward for current platform capabilities. It is important to understand that it is the responsibility of the platform to meet the processor's demand for memory bandwidth with an appropriate supply. The system bus of the Pentium 4 processor provides a high-bandwidth channel for this brokerage function of the platform.

For the large body of existing applications, memory bandwidth demanded by the application is relatively low, so performance improvements can be realized with processor scaling alone. However, new applications and benchmarks require that the platform capabilities meet higher requirements to deliver performance. The Pentium 4 processor with its high-performance microarchitecture is capable of generating high demands for memory bandwidth, should the application require it. To fulfill

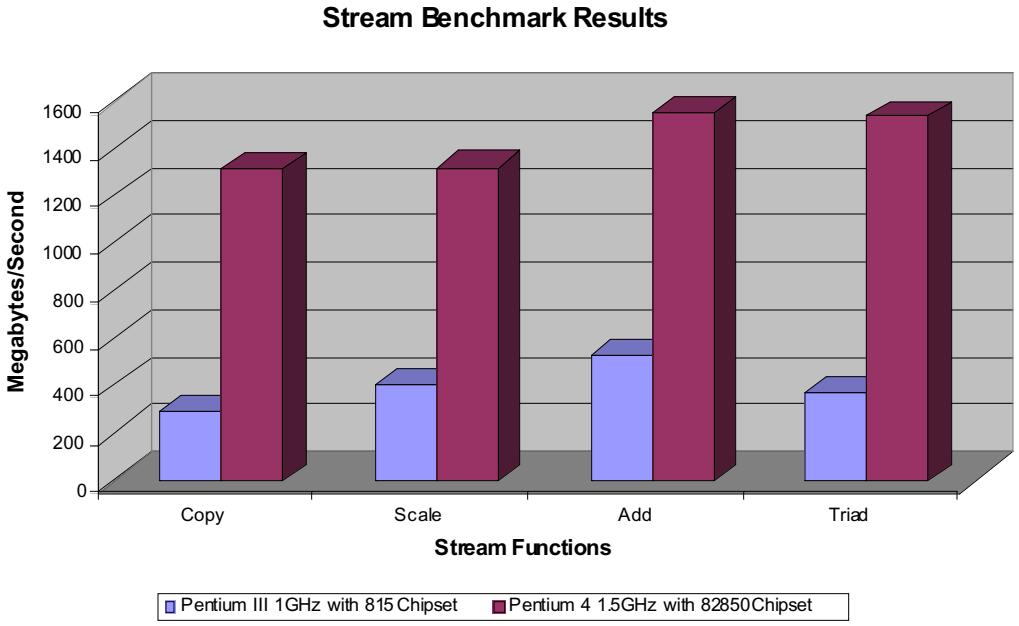

Figure 10: *STREAM* benchmark results

this demand, the platform components, primarily the system bus and memory connection, have been dramatically improved to supply this demand.

REFERENCES

1. McCalpin, John D., "Sustainable Memory Bandwidth in Current High-Performance Computers," October 12, 1995.
2. Hennessy, John L., and Patterson, David A., *Computer Architecture: A Quantitative Approach, Second Edition*, Morgan Kaufmann Publishers, ISBN: 1558603298.
3. SPEC CPU2000 information can be found at http://www.spec.org/osg/cpu2000/.
4. SYSmark 2000 information can be found at http://www.bapco.com/sysmark2000primer.htm.
5. Information on the 82850 Memory Controller hub can be downloaded from http://developer.intel.com/design/chipsets/datashts/290691.htm.

AUTHORS' BIOGRAPHIES

R. Scott Tetrick is responsible for CPU Platform Architecture in the Desktop Products Group. He has been involved with bus development at Intel since 1979. He holds ten patents on platform architecture. While at Intel, he has developed platforms from single board computer systems to supercomputers and embedded controllers to multiprocessing servers. His e-mail address is stetrick@ichips.intel.com.

Blaise Fanning is a Platform Architect in DPG's chipset engineering group. He received his B.S. and M.S. degrees in computer engineering from Boston University in 1987. He joined Intel in 1997 and was the architect of the Intel840 workstation chipset. He is currently responsible for platform performance issues and developing next-generation I/O interconnects. His e-mail address is blaise.fanning@intel.com.

Robert Greiner is a platform architect with the DPG Oregon Architecture team. He helped develop the quad pumped system bus for the Pentium® 4 processor. He also contributed to the performance simulators for the Pentium 4 processor. He has worked on high-speed, scalable interconnect protocols for Futurebus+, MIT, and others. He graduated with a B.S. degree in Mathematics from Michigan State University. His e-mail address is rgreiner@ichips.intel.com.

Tom Huff is an architect in the Intel® Architecture Group in Oregon. He was one of the architects in the core team that defined the Streaming SIMD Extensions for the IA-32 architecture. He also worked on multimedia performance analysis for the Pentium® 4 processor. He holds M.S. and Ph.D. degrees in Electrical Engineering from the University of Michigan. His e-mail address is thuff@ichips.intel.com.

Lance Hacking joined Intel's IA-32 Architecture Group in Oregon in 1994 after completing undergraduate studies

at Brigham Young University. His focus on multimedia performance began with the Pentium® Pro processor, includes the Streaming SIMD Extensions defininition, and continues today with the current Pentium® 4 processor. His e-mail address is Lance.Hacking@intel.com.

Dave L. Hill joined Intel's DPG team in 1993, and has been the bus cluster microarchitect on P4 Willamette and Foster projects. Dave has 20 years industry experience primarily in high-performance memory system microarchitecture, logic design, and system debug. His e-mail address is dlhill@ichips.intel.com.

Srinivas Chennupaty is a processor architect in the Desktop Products Group in Oregon. He was one of the architects in the core team that defined the Internet Streaming SIMD Extensions for the IA-32 architecture. He is currently working on multimedia performance analysis for the Pentium® 4 processor. He holds a M.S. degree in Computer Engineering from the University of Texas at Austin. His e-mail address is chennu@ichips.intel.com.

David A. Koufaty received B.S. and M.S. degrees from the Simon Bolivar University, Venezuela in 1988 and 1991, respectively. He then received a Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign in 1997. For the last three years he has worked for the DPG CPU Architecture organization. His main interests are in multiprocessor architecture and software, performance, and compilation. His e-mail address is dkoufaty@ichips.intel.com.

Subba Palacharla joined Intel in 1998. Since then he has been a member of the DPG Architecture team working on performance evaluation of Willamette and Foster systems. Subba graduated from the Indian Institute of Technology, Kharagpur, India in 1991 with a B.Tech degree in Computer Science and Engineering. Subba received his Ph.D. degree in Computer Science from the University of Wisconsin at Madison in 1998. His e-mail address is subbarao@ichips.intel.com.

Jeff Rabe is a chipset architect in the Desktop Product Group, and was the lead architect for the 82850 chipset. He joined Intel in 1980, and has worked as a Yield Analysis engineer, Product Engineer, and Application Engineer prior to his chipset architecture role. His e-mail address is jeff.l.rabe@intel.com.

Mike Derr is an architect working on the I/O Controller Hub (ICH) product line within the Desktop Platforms Group. He has eight years of experience in Intel® chipset development and holds four U.S. patents related to chipset technology. He received a B.S.E.E. degree from Tennessee Technological University and an M.S.E.E. degree from the Georgia Institute of Technology. His e-mail address is mike.n.derr@intel.com.